WHAT IS CLAIMED IS:

1. In a programmable computer, a method of merging an original data stream with a sequential plurality of delta streams to build an updated data stream, the method comprising the steps of:

a) initiating a search request, within the sequential plurality of delta streams, for a number of data bytes to transfer to the updated data stream;

b) fulfilling the search request with data bytes provided by the last delta stream in the sequential plurality of delta streams which is capable of supplying data bytes;

c) if the sequential plurality of delta streams is incapable of fulfilling the search request, fulfilling the search request with data bytes provided by the original data stream; and

d) repeating steps a)-c) until a search request cannot be fulfilled, and the updated data stream is complete.

2. A method as in claim 1, wherein:

a) each of the delta streams comprises a sequence of match and/or data delta frames;

b) the match frames describe matching segments of a delta stream and a prior stream in terms of byte addresses; and

c) the data frames comprises data in a delta stream which does not appear in a prior stream.

3. A method as in claim 1, wherein:

43

a) each of the delta streams comprises a sequence of match and/or data delta frames;

b) the match frames describe matching segments of a delta stream and a prior stream in terms of line addresses; and

c) the data frames comprise data in a delta stream which does not appear in a prior stream.

4. A method as in claim 1, further comprising the step of reading the original data stream directly from a sequential media.

5. A method as in claim 1, further comprising the step of writing the updated data stream to a sequential media.

6. In a programmable computer, a method of merging an original data stream with a plurality of delta streams to build an updated data stream, the method comprising the steps of:

a) constructing a chain of transaction elements corresponding to the sequential plurality of delta streams, wherein a highest numbered transaction element in the transaction chain is associated with a delta stream representing a latest revision to the original data stream, a lowest numbered transaction element in the transaction chain is associated with a delta stream representing a first revision to the original data stream, and consecutively numbered transaction elements in the transaction chain are associated with sequential revisions to the original data stream;

b) initiating a search request, within the transaction chain, for a number of data bytes to transfer to the updated data stream;

c) fulfilling the search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes;

d) if the transaction elements of the transaction chain are incapable of fulfilling the search request, fulfilling the search request with data bytes provided by the original data stream; and

e) repeating steps a)-d) until a search request cannot be fulfilled, and the updated data stream is complete.

7. A method as in claim 6, further comprising the step of reading the original data stream directly from a sequential media.

8. A method as in claim 6, further comprising the step of writing the updated data stream to a sequential media.

9. A method as in claim 6, further comprising the step of building a sequential plurality of negative delta streams corresponding to both the transaction elements of the transaction chain and the sequential plurality of delta streams.

10. A method as in claim 6, wherein:

a) each transaction element in the transaction chain sequentially loads a sequence of match and/or data delta frames from its corresponding delta stream; and

b) the search request initiated within the transaction chain is initiated within loaded delta frames of the transaction elements.

11. A method as in claim 10, wherein:

a) a match frame comprises,

i) a beginning byte address for a segment of data in a delta stream which matches a segment of data in a prior stream; and

ii) a source match position indicating a beginning byte address in the prior stream which corresponds to the beginning byte address in the delta stream; and

b) a data frame comprises data in the delta stream which does not appear in the prior stream.

12. A method as in claim 11, further comprising the step of building a sequential plurality of negative delta streams corresponding to both the transaction elements of the transaction chain and the sequential plurality of delta streams.

13. A method as in claim 12, wherein the building of a particular negative delta stream in the sequential plurality of negative delta streams comprises the steps of:

a) monitoring a prior stream position associated with a particular transaction element;

b) when a match frame is loaded by the particular transaction element, calculating a prior stream lag count as the difference between the loaded match frame's source match position and the transaction element's prior stream position;

c) transferring a number of data bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element;

d) creating an inverse of the loaded match frame; and

e) recording the inverse match frame in the particular negative delta stream corresponding to the particular transaction element.

14. A method as in claim 13, wherein the negative delta streams are attached to null devices.

15. A method as in claim 13, wherein the step of transferring a number of data bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element, comprises the steps of:

a) initiating a second search request, within the loaded delta frames of those transaction elements of the transaction chain which are prior to the particular transaction element, for a number of data bytes, equaling the prior stream lag count, to transfer, as a data frame, to the particular negative delta stream corresponding to the particular transaction element;

b) fulfilling the second search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes;

c) if the transaction elements of the transaction chain prior to the particular transaction element are incapable of fulfilling the second search request, fulfilling the second search request with data bytes provided by the original data stream;

d) decrementing the prior stream lag count by the number of data bytes provided in fulfilling the second search request; and

e) repeating steps a)-d) until the prior stream lag count is equal to zero.

16. A method as in claim 11, wherein a loaded delta frame of a transaction element is the transaction element's current frame, and the method comprises the further steps of:

a) monitoring a number of unused bytes in the current frame of each of the transaction elements;

b) loading a next delta frame of a delta stream into the current frame of its corresponding transaction element whenever the number of unused bytes in the transaction element's current frame is determined to be zero; and

c) after the above step, resetting the number of unused bytes in the current frame of the transaction element.

17. A method as in claim 16, wherein:

a) the step of initiating a search request comprises initiating the search request within the current frame of the highest numbered transaction element in the transaction chain;

b) the step of fulfilling the search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes, comprises the steps of:

i) if the current frame contains fewer than the requested number of data bytes, updating the requested number of data bytes to reflect a number of data bytes in the current frame;

ii) if the current frame is a data frame, fulfilling the search request by transferring the requested number of data bytes from the current frame to the updated data stream;

iii) if the current frame is a match frame, and a prior delta stream exists,

A) passing the search request to the next lower numbered transaction element in the transaction chain; and

B) repeating the method beginning with step b)i) of this claim; and the method comprises the further step of,

c) after fulfilling the search request, and for each of the transaction elements to which the search request was passed, decrementing the number of unused bytes in each of the transaction element's current frames, and, but for the transaction element which fulfilled the search request, incrementing a prior stream position of each of the transaction elements, by the number of data bytes transferred to the updated data stream during the fulfillment of the search request.

18. A method as in claim 17, further comprising the step of reading the original data stream directly from a sequential media.

19. A method as in claim 17, further comprising the step of writing the updated data stream to a sequential media.

20. A method as in claim 17, further comprising the step of building a sequential plurality of negative delta streams corresponding to both the transaction elements of the transaction chain and the sequential plurality of delta streams.

21. A method as in claim 20, wherein the building of a particular negative delta stream in the sequential plurality of negative delta streams comprises the steps of:

a) monitoring the prior stream position associated with a particular transaction element corresponding to the particular negative delta stream;

b) when a match frame is loaded by the particular transaction element, calculating a prior stream lag count as the difference between the match frame's source match position and the transaction element's prior stream position;

c) transferring a number of data bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element;

d) creating an inverse of the loaded match frame; and

e) recording the inverse match frame in the particular negative delta stream corresponding to the particular transaction element.

22. A method as in claim 21, wherein the step of transferring a number of data bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element, comprises the steps of:

a) initiating a second search request, within the loaded delta frames of those transaction elements of the transaction chain which are prior to the particular transaction element, for a number of data bytes, equaling the prior stream lag count, to transfer, as a data frame, to the particular negative delta stream corresponding to the particular transaction element;

b) fulfilling the second search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes;

c) if the transaction elements of the transaction chain prior to the particular transaction element are incapable of fulfilling the second search request, fulfilling the second search request with data bytes provided by the original data stream;

d) decrementing the prior stream lag count by the number of data bytes provided in fulfilling the second search request; and

e) repeating steps a)-d) until the prior stream lag count is equal to zero.

23. A method as in claim 22, wherein:

a) the step of initiating a second search request comprises initiating a second search request within the current frame of the first transaction element in the transaction chain which is prior to the particular transaction element;

b) the step of fulfilling the second search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes, comprises the steps of:

i) if the current frame contains fewer than the prior stream lag count number of data bytes, updating the second requested number of data bytes to reflect a number of data bytes in the current frame;

ii) if the current frame is a data frame, fulfilling the second search request by transferring the second requested number of data bytes from the current frame to the particular negative delta stream corresponding to the particular transaction element;

iii) if the current frame is a match frame, and a prior delta stream exists,

A) passing the second search request to the next lower numbered transaction element in the transaction chain; and

B) repeating the method beginning with step b)i) of this claim; and the method comprises the further step of,

c) after fulfilling the second search request,

i) for each of the transaction elements in the transaction chain to which the second search request was passed, decrementing the number of unused bytes in each of the transaction element's current frames by the number of data bytes transferred to the particular negative delta stream corresponding to the particular transaction element during the fulfillment of the second search request; and

ii) for each of the transaction elements having a numerical position in the transaction chain which is greater than the numerical position of the transaction element which fulfilled the second search request, but less than or equal to the numerical position of the particular transaction element, incrementing each of the transaction element's prior stream positions by the number of data bytes transferred to the particular negative delta stream corresponding to the particular transaction element during the fulfillment of the second search request.

24. In a programmable computer, a method of merging an updated data stream with one or more negative delta streams to reconstruct a desired prior data stream, the method comprising the steps of:

a) constructing a chain of transaction elements corresponding to the plurality of negative delta streams, wherein a highest numbered transaction element in the transaction chain is associated with a negative delta stream representing a last required revision to the updated data stream, a lowest numbered transaction element in the transaction chain is associated with a negative delta stream representing a first required revision to the updated data stream, and consecutively numbered transaction elements in the transaction chain are associated with sequentially required revisions to the updated data stream;

b) initiating a search request, within the transaction chain, for a number of data bytes to transfer to the desired prior data stream;

c) fulfilling the search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes;

d) if the transaction elements of the transaction chain are incapable of fulfilling the search request, fulfilling the search request with data bytes provided by the updated data stream; and

e) repeating steps a)-d) until a search request cannot be fulfilled, and the desired prior data stream is complete.

25. In a programmable computer, a method of constructing a negative delta stream during a merge of an original data stream with a delta stream, the method comprising the steps of:

a) initiating a search request, within a current match and/or data delta frame of the delta stream, for a number of data bytes to transfer to an updated data stream;

b) if the current delta frame of the delta stream is empty, loading a next delta frame from the delta stream as the current delta frame;

c) monitoring an original data stream position associated with the delta stream;

d) after loading a next delta frame which is a match frame,

i) calculating a prior stream lag count as the difference between the match frame's source match position and the delta stream's original data stream position;

ii) transferring a number of bytes, equaling the prior stream lag count, from the original data stream to the negative delta stream, in the form of a data frame;

iii) creating an inverse of the match frame;

iv) recording the inverse match frame in the negative delta stream; and

v) incrementing the original data stream position associated with the delta stream by a number equal to the prior stream lag count;

e) if the delta stream is capable of supplying data bytes from a current frame which is a data frame, fulfilling the search request with data bytes provided from the current frame;

f) if the delta stream is incapable of fulfilling the search request, fulfilling the search request with data bytes provided by the original data stream;

g) incrementing the original data stream position associated with the delta stream by the number of data bytes provided to the updated data stream by the original data stream; and

h) repeating steps a)-g) until a search request cannot be fulfilled, and the updated data stream is complete.

26. A method as in claim 25, further comprising the step of writing the negative delta stream to a sequential media.

27. In a programmable computer, a method of merging a sequential plurality of delta streams to create a compiled delta stream, the method comprising the steps of:

a) initiating a search request, within the sequential plurality of delta streams, for a data frame to transfer to the compiled delta stream;

b) fulfilling the search request with a data frame provided by the last delta stream in the sequential plurality of delta streams which is capable of supplying a data frame;

c) if the sequential plurality of delta streams is incapable of fulfilling the search request, constructing a match frame to transfer to the compiled delta stream; and

d) repeating steps a)-c) until a search request cannot be fulfilled, and the compiled delta stream is complete.

28. A method as in claim 27, wherein:

a) each of the delta streams comprises a sequence of match and/or data delta frames;

b) the match frames describe matching segments of a delta stream and a prior stream in terms of byte addresses; and

c) the data frames comprise data in a delta stream which does not appear in a prior stream.

29. A method as in claim 27, wherein:

a) each of the delta streams comprises a sequence of match and/or data delta frames;

b) the match frames describe matching segments of a delta stream and a prior stream in terms of line addresses; and

c) the data frames comprise data in a delta stream which does not appear in a prior stream.

30. A method as in claim 27, further comprising the step of reading the sequential plurality of delta streams directly from a sequential media.

31. A method as in claim 27, further comprising the step of writing the compiled delta stream to a sequential media.

32. In a programmable computer, a method of merging a sequential plurality of delta streams to create a compiled delta stream, the method comprising the steps of:

a) constructing a chain of transaction elements corresponding to the sequential plurality of delta streams, wherein a highest numbered transaction element in the transaction chain is associated with a delta stream representing a latest revision to an original data stream, a lowest numbered transaction element in the transaction chain is associated with a delta stream representing a first revision to the original data stream, and consecutively numbered transaction elements in the transaction chain are associated with sequential revisions to the original data stream;

b) initiating a search request, within the transaction chain, for a data frame to transfer to the compiled delta stream;

c) fulfilling the search request with a data frame provided by the highest numbered transaction element in the transaction chain which is capable of supplying a data frame;

d) if the transaction elements of the transaction chain are incapable of fulfilling the search request, constructing a match frame to transfer to the compiled delta stream; and

e) repeating steps a)-d) until a search request cannot be fulfilled, and the compiled delta stream is complete.

33. A method as in claim 32, further comprising the step of reading the sequential plurality of delta streams directly from a sequential media.

34. A method as in claim 32, further comprising the step of writing the compiled delta stream to a sequential media.

35. A method as in claim 32, wherein:

a) each transaction element in the transaction chain sequentially loads a sequence of match and/or data delta frames from its corresponding delta stream; and

b) the search request initiated within the transaction chain is initiated within loaded delta frames of the transaction elements.

36. A method as in claim 35, wherein:

a) a match frame comprises,

i) a beginning byte address for a segment of data in a delta stream which matches a segment of data in a prior stream; and

ii) a source match position indicating a beginning byte address in the prior stream which corresponds to the beginning byte address in the delta stream; and

b) a data frame comprises data in the delta stream which does not appear in the prior stream.

37. A method as in claim 36, further comprising the step of building a sequential plurality of negative delta streams corresponding to both the transaction elements of the transaction chain and the sequential plurality of delta streams.

38. A method as in claim 37, wherein the building of a particular negative delta stream in the sequential plurality of negative delta streams, comprises the steps of:

a) monitoring a prior stream position associated with a particular element in the transaction chain;

c) when a match frame is loaded by the particular transaction element, calculating a prior stream lag count as the difference between the match frame's source match position and the transaction element's prior stream position;

d) transferring representations of a number of bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element;

e) creating an inverse of the loaded match frame; and

f) recording the inverse match frame in the particular negative delta stream corresponding to the particular transaction element.

39. A method as in claim 38, wherein the negative delta streams are attached to null devices.

40. A method as in claim 38, wherein the step of transferring representations of a number of bytes equaling the prior stream lag count, from streams prior to the particular transaction element, to the particular negative delta stream corresponding to the particular transaction element, comprises the steps of:

a) initiating a second search request, within the loaded delta frames of those transaction elements of the transaction chain which are prior to the particular transaction element for a number of data bytes, equaling the prior stream lag count, to transfer to a data frame of the particular negative delta stream associated with the particular transaction element;

b) fulfilling the second search request with data bytes provided by the highest numbered transaction element in the transaction chain capable of supplying data bytes;

c) if the transaction elements of the transaction chain prior to the particular transaction element are incapable of fulfilling the second search request, fulfilling the second search request with a match frame referencing the original data stream;

d) decrementing the prior stream lag count by the number of data bytes provided in fulfilling the second search request; and

e) repeating steps a)-d) until the prior stream lag count is equal to zero.

41. A computer programmed to merge a sequential plurality of delta streams with an original data stream, in a single pass, to create an updated data stream, the computer comprising:

a) a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a delta stream input corresponding to one of the sequential plurality of delta streams; and

b) a consumer process, connected to a trailing element of the transaction chain, and comprising an input for the original data stream and an output for the updated data stream.

42. A computer as in claim 41, wherein each of the transaction elements further comprise:

a) a delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface.

43. A computer as in claim 42, wherein each of the transaction elements further comprises a negative delta stream writer.

44. A computer programmed to merge a sequential plurality of negative delta streams with an updated data stream, in a single pass, to reconstruct a desired prior data stream, the computer comprising:

a) a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a negative delta stream input corresponding to one of the sequential plurality of negative delta streams; and

b) a consumer process, connected to a trailing element of the transaction chain, and comprising an input for the updated data stream and an output for the desired prior data stream.

45. A computer as in claim 44, wherein each of the transaction elements further comprises:

a) a negative delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface.

46. A computer programmed to merge a sequential plurality

of delta streams, in a single pass, to create a compiled delta stream, the computer comprising:

a) a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a delta stream input corresponding to one of the sequential plurality of delta streams; and

b) a compiler consumer process, connected to a trailing element of the transaction chain, comprising an output for the compiled delta stream.

47. A computer as in claim 46, wherein each of the transaction elements further comprises:

a) a delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface.

48. A computer as in claim 47, wherein each of the transaction elements further comprises a negative delta stream writer.

49. A physical storage media programmed to control a computer in merging a sequential plurality of delta streams with an original data stream, in a single pass, to create an updated data stream, the media comprising:

a) means to produce a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a delta

64

stream input corresponding to one of the sequential plurality of delta streams;

b) a consumer process comprising an input for the original data stream and an output for the updated data stream; and

c) means to connect the consumer process to a trailing element of the transaction chain.

50. A physical storage media as in claim 49, wherein the means to produce a transaction chain comprises means to produce:

a) a delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface, for each of the transaction elements.

51. A physical storage media as in claim 50, wherein the means to produce a transaction chain comprises means to produce a negative delta stream writer for each of the transaction elements.

52. A physical storage media programmed to control a computer in merging a sequential plurality of negative delta streams with an updated data stream, in a single pass, to create a desired prior data stream, the media comprising:

a) means to produce a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a negative delta stream input corresponding to one of the sequential plurality of negative delta streams;

b) a consumer process comprising an input for the updated data stream and an output for the previous data stream; and

c) means to connect the consumer process to a trailing element of the transaction chain.

53. A physical storage media as in claim 52, wherein the means to produce a transaction chain comprises means to produce:

a) a negative delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface, for each of the transaction elements.

54. A physical storage media programmed to control a computer in merging a sequential plurality of delta streams, in a single pass, to create a compiled delta stream, the media comprising:

a) means to produce a transaction chain comprising a plurality of sequenced transaction elements, wherein each transaction element has a delta stream input corresponding to one of the sequential plurality of delta streams;

b) a compiler consumer process comprising an output for the compiled delta stream; and

c) means to connect the compiler consumer process to a trailing element of the transaction chain.

55. A physical storage media as in claim 54, wherein the

means to produce a transaction chain comprises means to produce:

a) a delta stream reader;

b) a prior transaction element index;

c) a prior stream position monitor;

d) a current frame buffer; and

e) a search request fulfillment I/O interface, for each of the transaction elements.

56. A physical storage media as in claim 55, wherein the means to produce a transaction chain comprises means to produce a negative delta stream writer for each of the transaction elements.